

Programação Orientada a Objetos

Exercício – Tratamento de Erros e Exceções

Exercício – Erros e Exceções

- 1) Crie uma classe *CalculoMatematico*
 - Nela, crie um método *divisao*, que recebe como parâmetros os valores a serem divididos. O retorno é o resultado da divisão (todos os números devem ser do tipo inteiro). Crie uma classe de teste para testar *calculoMatematico*
 - Nela crie um objeto *calculoMatematico* e aceda ao método *divisao*, tentando dividir 4 por 0.
- Execute a classe e veja o que acontece.

Exercício – Erros e Exceções

- 2) Crie um bloco *try...catch* no método *divisao* para tratar a operação realizada *catch*:
 - a) Informar o objeto do tipo *ArithmeticException*
 - b) Imprimir uma mensagem informando que a operação não pode ser realizada
 - c) Retorna zero

Exercício – Erros e Exceções

- 3) Tire o bloco *try...catch* do método *divisao*
 - a) Adicione *throws ArithmeticException* na assinatura do método
 - b) E na primeira linha do bloco do método, faça uma verificação se o divisor é igual a 0
 - c) Se for, lance uma exceção:
 - *throw new ArithmeticException("Texto");*

Exercício – Erros e Exceções

- 3) Tire o bloco *try...catch* do método *divisao*
 - d) Na classe de teste, crie um bloco *try...catch*, tentando executar o método divisão *catch* para *ArithmeticException*
 - e) No bloco do *catch*, imprima o método *getMessage()* do objeto criado do tipo *ArithmeticException*

Exercício – Erros e Exceções

- 4) Crie uma nova classe *DivisaoPorZeroException*
- Na classe *CalculoMatematico*, troque *ArithmeticException* por *DivisaoPorZeroException*
 - a) *throws DivisaoPorZeroException*
 - b) *throw new DivisaoPorZeroException();*
 - c) Na classe de teste, troque no catch *ArithmeticException* por *DivisaoPorZeroException*

P O O - Exercício

Questões?